This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (Previously Presented) A business process service debugger for remotely debugging a business process service, comprising:

means for establishing a communications connection with a remote computer, wherein the remote computer is implementing the business process service;

means for reading stored state information regarding events related to at least one business process implemented for the business process service;

means for displaying a symbolic representation of the operation of the business process service based on the stored state information; and

means for remotely debugging the business process service using the symbolic representation, communications connection and stored state information.

2. (Previously Presented) The business process service debugger of claim 1, wherein business processes and instances of the business process service other than those being debugged are not disrupted during debugging.

3. (Previously Presented) The business process service debugger of claim 1, wherein the symbolic representation comprises a workflow of at least one business process in the business process service.

4. (Previously Presented) The business process service debugger of claim 1, further comprising means for interacting with the business process service according to a user instruction.

5. (Original) The business process service debugger of claim 1, wherein the stored state information corresponds to a variable assignment within the business process service.

6. (Previously Presented) The business process service debugger of claim 1, wherein the events are historical events that occurred prior to failure of the at least one business process.

7. (Previously Presented) The business process service debugger of claim 1, wherein the stored state information corresponds to message flow data and an order in which run time components performed the one business process as a result of message processing.

8. (Original) The business process service debugger of claim 1, wherein said reading means further comprises means for reading stored business process service configuration information.

9. (Previously Presented) The business process service debugger of claim 1, wherein the events are events that occur prior to an inserted breakpoint in the one business process.

10. (Original) The business process service debugger of claim 1, wherein said debugging means comprises means for detecting a location where the instance is being processed.

11. (Original) The business process service debugger of claim 1, wherein said debugging means comprises means for detecting a location where the instance state is being stored.

12. (Currently Amended) A system for remotely debugging a distributed transactional application, comprising:

a server configured to execute an instance of a business process service comprising a plurality of business processes, thereby generating runtime data;

a client computer configured to execute a debugging user interface (UI) process that establishes a communications connection with the server requests runtime data for at least one of the plurality of business processes, and generates, based on the runtime data, a symbolic

representation of the business service process showing any debugging break points specified by a user; and

an interceptor for monitoring the runtime data and, when a specified break point is identified, causing the server to enter or leave a debugging state.

13. (Original) The system of claim 12, further comprising a database for receiving the runtime data and for storing business process service state information.

14. (Previously Presented) The system of claim 13, further comprising a display device for displaying the symbolic representation, and a user input device, wherein the input device is used to specify debugging break points.

15. (Previously Presented) The system of claim 14, wherein the symbolic representation comprises a workflow representative of the program flow of the business process service.

16. (Previously Presented) The system of claim 14, wherein the display device further displays data representative of a message flow of the business process service.

17. (Previously Presented) The system of claim 14, wherein the symbolic representation is presented according to stored state information.

18. (Previously Presented) The system of claim 12, wherein a message box database is coupled between the server and client computer and is configured for communicating debugging requests from the client computer.

19. (Previously Presented) The system of claim 18, wherein the UI process comprises an application program interface for communicating with the message box database.

20. (Previously Presented) The system of claim 18, further comprising a tracking database to receive business process service tracking information, wherein the UI process comprises a UI component for communicating with the tracking database.

21. (Canceled)

22. (Canceled)

23. (Original) The system of claim 12, wherein the interceptor is a component of a computer language that provides stored state tracking information.

24. (Original) The system of claim 12, wherein the UI process detects a location where the instance is being processed.

25. (Original) The system of claim 12, wherein the UI process detects a location where the instance state is being stored.

26. (Previously Presented) A method for debugging an instance of a business process service running on a remote computer, comprising:

generating for display, in a graphical user interface (GUI), a symbolic representation of the business process service based on a correlation of information about the design and execution of the business process service;

receiving a debugging command generated by a user interacting with the GUI;

establishing a direct client connection channel with the remote computer;

causing an interceptor to monitor runtime data generated by the instance of the business process service in accordance with the debugging command;

receiving a runtime request, generated by a user interacting with the GUI, for event information generated by execution of the instance of the business process service;

sending the runtime request to the remote computer for processing by the remote computer.

27. (Original) The method of claim 26, further comprising:

querying a database containing a status of the business process service;

displaying a query result on a display device;

receiving user input with respect to the query result; and

establishing the direct client connection channel in response to the user input.

28. (Original) The method of claim 27, wherein the information contained in the database is instance runtime data.

29. (Original) The method of claim 27, wherein the information contained in the database is instance tracking data.

30. (Previously Presented) The method of claim 26, further comprising:

creating the business process service using a process designer;

saving a business process service configuration and symbolic data in a database as information about the design of the business process service;

displaying the symbolic representation on a display device according to the saved business process service configuration and symbolic data;

generating a runtime request based on the symbolic representation; and

displaying a result of the runtime request on the display device.

31. (Previously Presented) The method of claim 30, wherein the symbolic representation comprises a shape corresponding to an operation in the business process service.

32. (Previously Presented) The method of claim 30, wherein the symbolic representation comprises a workflow representation of the business process service.

33. (Original) The method of claim 30, wherein the saving step takes place in connection with compiling and deploying the business process service.

34. (Original) The method of claim 30, wherein the business process service is implemented in a computer language that provides stored state information.

35. (Previously Presented) The method of claim 26, wherein the debugging command is a break point.

36. (Canceled)

37. (Previously Presented) The method of claim 26, wherein the runtime data is state information.

38. (Original) The method of claim 26, further comprising detecting a location where the instance is being processed.

39. (Original) The method of claim 26, further comprising detecting a location where an instance state is being stored.

40. (Previously Presented) A method in a computer system for displaying on a display device a graphical debugging environment for a business process service, the method comprising:

obtaining design information about the business process service;

obtaining tracking information about execution of the business process service;

generating a symbolic representation of the operation of the business process service according to the design information and tracking information; and

displaying on the display device a graphical debugging environment showing the symbolic representation.

41. (Original) The method of claim 40, further comprising receiving runtime data for the business process service and presenting the runtime data on the display device.

42. (Previously Presented) The method of claim 41, wherein the runtime data comprises historical message flow information including identification of run time messages that were constructed as a result of processing received messages, and further comprises order information pertaining to the order in which different run time components were executed as a result of processing received messages.

43. (Previously Presented) The method of claim 40, wherein the graphical debugging environment enables a user to place a breakpoint in the symbolic representation of the operation of the business process service.

44. (Previously Presented) The method of claim 40, the symbolic representation comprising symbols, wherein the graphical debugging environment also displays information about the symbols.

45. (Previously Presented) The method of claim 40, further comprising receiving input from an input device to place a break point proximate a symbol, and presenting a symbol representing the break point on the symbolic representation.

46. (Previously Presented) A computer-readable storage medium having computer-executable instructions for performing a method for debugging an instance of a business process service running on a remote computer, comprising:

generating for display, in a graphical user interface (GUI), a symbolic representation of the business process service based on a correlation of information about the design and execution of the business process service;

receiving a debugging command generated by a user interacting with the GUI;

establishing a direct client connection channel with the remote computer;

causing an interceptor to monitor runtime data generated by the instance of the business process service in accordance with the debugging command;

receiving a runtime request; and

sending the runtime request to the remote computer for processing by the remote computer.

47. (Previously Presented) The computer-readable storage medium of claim 46, wherein the method further comprises:

querying a database containing a status of the business process service;

displaying a query result on a display device;

receiving user input with respect to the query result; and

establishing the direct client connection channel in response to the user input.

48. (Previously Presented) The computer-readable storage medium of claim 47, wherein the information contained in the database is instance runtime data.

49. (Previously Presented) The computer-readable storage medium of claim 47, wherein the information contained in the database is instance tracking data.

50. (Previously Presented) The computer-readable storage medium of claim 46, wherein the method further comprises:

creating the business process service using a process designer;

saving business process service configuration data in a database as information about the design of the business process service;

displaying the symbolic representation on a display device according to the saved business process service configuration data;

generating a runtime request based on the symbolic representation; and

displaying a result of the runtime request on the display device.

51. (Previously Presented) The computer-readable storage medium of claim 50, wherein the symbolic representation comprises a shape corresponding to an operation in the business process service.

52. (Previously Presented) The computer-readable storage medium of claim 50, wherein the symbolic comprises a workflow representation of the business process service.

53. (Previously Presented) The computer-readable storage medium of claim 50, wherein the saving step takes place in connection with compiling and deploying the business process service.

54. (Previously Presented) The computer-readable storage medium of claim 50, wherein the business process service is implemented in a computer language that provides stored state information.

55. (Previously Presented) The computer-readable storage medium of claim 50, wherein the debugging command is a break point.

56. (Previously Presented) The computer-readable storage medium of claim 50, wherein the debugging command is a request for data regarding an instance of the business process service.

57. (Previously Presented) The computer-readable storage medium of claim 56, wherein the runtime data is state information.

58. (Previously Presented) The computer-readable storage medium of claim 46, wherein the method further comprises detecting a location where the instance is being processed.

59. (Previously Presented) The computer-readable storage medium of claim 46, wherein the method further comprises detecting a location where an instance state is being stored.

60. (Previously Presented) A computer-readable storage medium having computer-executable instructions for performing a method for displaying on a display device a graphical debugging environment for a business process service, the method comprising:

obtaining design information about the business process service;

obtaining configuration information about the business process service;

generating a symbolic representation of the operation of the business process service according to the design information and configuration information; and

displaying on the display device a graphical debugging environment showing the symbolic representation.

61. (Previously Presented) The computer-readable storage medium of claim 60, wherein the method further comprises receiving runtime data for the business process service and presenting the runtime data on the display device.

62. (Previously Presented) The computer-readable storage medium of claim 61, wherein the runtime data comprises message flow information.

63. (Previously Presented) The computer-readable storage medium of claim 60, wherein the graphical debugging environment enables a user to place a breakpoint in the symbolic representation of the operation of the business process service.

64. (Previously Presented) The computer-readable storage medium of claim 60, the symbolic representation comprising symbols, wherein the graphical debugging environment also displays information about the symbols.

65. (Previously Presented) The computer-readable storage medium of claim 60, wherein the method further comprises receiving input from an input device to place a break point proximate a symbol, and presenting a symbol representing the break point on the symbolic representation.